

[First Hit](#) [Fwd Refs](#)[Previous Doc](#) [Next Doc](#) [Go to Doc#](#) [Generate Collection](#) [Print](#)

L4: Entry 6 of 8

File: USPT

Apr 3, 2001

DOCUMENT-IDENTIFIER: US 6212524 B1

TITLE: Method and apparatus for creating and populating a datamart

Brief Summary Text (8):

The advantage of datamarts is that users can quickly access data that is important to their business decision making. To meet this goal, datamarts should have the following characteristics. First, datamarts should be consistent in that they give the same results for the same search. The datamart should also be consistent in the use of terms to describe fields in the datamart. For example, "sales" has a specific definition, that when fetched from a database, provides a consistent answer. Datamarts should also be able to separate and combine every possible measure in the business. Many of these issues are discussed in the following book, Ralph Kimball, The Data Warehouse Toolkit, John Wiley and Sons, Inc., New York, N.Y. (1996).

Detailed Description Text (67):

Focusing on the datamart creation, the system allows a consultant to build a datamart from a schema definition and a definition of the sources of the data. From the schema definition, the system automatically builds the tables needed in the datamart. Also, from the schema definition, and the sources definition, the system can automatically extract the data from those sources. Depending on the semantic meaning of the data, as defined by the schema definition, the system automatically converts the data from the sources into forms that are readily usable in the datamart. Once the datamart has been created, and the data has been loaded, users can then perform queries on the data.

Detailed Description Text (104):

Once the staging tables 130 have been loaded, the semantic definitions 163 can be accessed from the enterprise manager 102 to convert the information in the staging tables 130 to predefined data semantics. These predefined data semantics allow for powerful queries, consistency in the definition of the meaning of the data in the datamart 150, and allow for changes to be made to the schema. Generally, the semantic template conversion 140 takes data stored in the staging tables 130, performs a conversion of that data according to a corresponding semantic definition (defined in the schema definitions 161), and populates the datamart 150 with the converted data.

Detailed Description Text (177):

Generally, a consultant will create a new datamart 150 by defining instances of the dimension bases 306, and constellations 302. Each instance corresponds to a row in the dimensions bases 306 table or the constellation 302 table. The constellation instances are defined by defining aggregates, dimensions, facts, measures, and ticksheets. The following describes the definition of a schema using the metadata 160. This corresponds to block 210 of FIG. 2.

Detailed Description Text (198):

The connector time stamp 407 relates to information about incremental extraction. An incremental extraction is where increments of the data in the source system 110 are extracted. The connector time stamp includes a connector key, a connector time stamp key, current max date, a current max time stamp, a last max date, and a last max time stamp. The connector key points to the connector to which the connector time stamp applies. The connector time stamp key is a primary key. The current max date is an indicator of the proposed new system date of the last successful extraction. The current maximum time stamp is the proposed new SQL server time stamp field for the last successful extraction. The last maximum date is the system date of the

last successful extraction. The last maximum time stamp is the SQL server time stamp field for the source system databases at the last successful extraction.

Detailed Description Text (200):

The connector column latch 409 defines information about incremental extraction based on a database column. The incremental extraction information is thus kept in the database and can be retrieved. The connector column latch 409 includes the following attributes: a column name, a connector column latch key, a connector key, a current maximum value, a last maximum value, and a table name. The table name is the name in the input data store for the corresponding connector. The column name is the column name within that table. The connector column latch key is the primary key. The connector key points to the connector to which this latch applies. The current max value represents the proposed new maximum value for the incremental extraction. This number is pushed into the last maximum value if the currently executing extraction succeeds. The last maximum value is the maximum value that was extracted during the last run of the extraction.

Detailed Description Text (215):

The semantic type 430 defines a set of predetermined semantic types for use in defining a schema. The semantic type includes a logical name for a particular transformation. Associated with the semantic type are a dimension semantic type 432 and a fact semantic type 434. The dimension semantic type table 432 defines the ways in which dimension data in the staging tables 130 can be extracted and put into the datamart 150. Similarly, the fact semantic type defines the ways in which the information in the staging tables 130 can be put into the fact tables of the datamart 150. Both the fact semantic type 434 and the dimension semantic type 432 include pointers to an actual table type and are used to subset the full list of semantic types.

Detailed Description Text (226):

The SQL server store table 456 defines details about an SQL server system. The SQL server store includes the following attributes: a data store key, a database name, a password, a server, and SQL server store key, a user name, and a version. The data store key is a one to one relationship to a data store entry. The database name is an SQL server database name (\$\$DEFAULT means the database in which this role resides). The password is the SQL server password. \$\$ DEFAULT again means the password currently logged into to read this data. The server is the SQL server name. The SQL server store key is the primary key. The user name is the SQL server user name. The version is the vendor's version number of this SQL server. \$\$DEFAULT means use the default value for the current database being used. For example, the database name means the database in which this role resides.

Detailed Description Text (233):

An SQL statement is a single step in an extraction run that represents a data push or a data pull. The SQL source code dictates the action for a given extraction node. After the SQL statements are run, the staging tables 130 are ready. The semantic conversion of the data in the staging tables 130 can occur.

Detailed Description Text (238):

Some embodiments of the invention correspond to only one or more semantic templates and a computer readable media, a computer, an electromagnetic waveform, or the like.

Detailed Description Text (262):

The schema modification involves comparing the changed schema definition with the present schema definition. As will be seen below, an actual table 502 keeps track of all of the dimension tables and the fact tables in the datamart 150. When a change is made to the schema definition, a comparison is made between the old definition and the new definition. The difference between these definitions defines the set of tables, columns, and rows that need to be added, deleted or modified, in some way. Importantly, the modifications can often be made without losing any data in the datamart 150.

Detailed Description Text (326):

The following describes a constellation used in a business. In this example a new dimension is

added very simply and the changes are automatically propagated into the datamart 150. The enterprise manager interface 192 is used by the consultant to define and manipulate the system 100.

Detailed Description Text (349):

In FIG. 24, returning to the connector steps window, we have switched to an All Semantics connector 2410. The All Semantics connector 2410 causes the semantic conversion of the information in the staging table for use in the datamart 150.

Detailed Description Text (352):

FIG. 27 illustrates the results of a consultant adding a new dimension 2700 (called warehouse) to the sales constellation 720. The batch operation window 1600 illustrates the changes that are being made to the datamart that was created in FIG. 16. To achieve these results, the consultant need only perform the following steps:

Detailed Description Text (353):

1. Define the new dimension.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

Hit List

[Clear](#)[Generate Collection](#)[Print](#)[Fwd Refs](#)[Bkwd Refs](#)[Generate OACS](#)

Search Results - Record(s) 1 through 8 of 8 returned.

1. Document ID: US 6772167 B1

L4: Entry 1 of 8

File: USPT

Aug 3, 2004

US-PAT-NO: 6772167

DOCUMENT-IDENTIFIER: US 6772167 B1

TITLE: System and method for providing a role table GUI via company group

DATE-ISSUED: August 3, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Snavely; Amy J.	Binghamton	NY		
Sjostrom; William M.	Rome	PA		
Musa; Mark A.	Brackney	PA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 09/ 656803 [PALM]

DATE FILED: September 7, 2000

PARENT-CASE:

CROSS REFERENCES TO RELATED APPLICATIONS The following U.S. patent applications filed concurrently herewith are assigned to the same assignee hereof and contain subject matter related, in certain respects, to the subject matter of the present application, the teachings of which applications are incorporated herein by this reference: Ser. No. 09/657,215, entitled "System and Method for Clustering Servers for Performance and Load Balancing", Ser. No. 09/657,216, entitled "System and Method for Front End Business Logic and Validation", Ser. No. 09/657,217, entitled "System and Method for Data Transfer With Respect to External Applications", Ser. No. 09/656,037, entitled "System and Method for Providing a Relational Database Backend", Ser. No. 09/656,967, entitled "System and Method for Populating HTML Forms Using Relational Database Agents", Ser. No. 09/657,196, entitled "System and Method for Catalog Administration Using Supplier Provided Flat Files", Ser. No. 09/657,195, entitled "System and Method for Providing an Application Navigator Client Menu Side Bar",

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/102; 707/3, 707/4, 707/9, 707/101, 707/103, 707/104

US-CL-CURRENT: 707/102; 707/101, 707/3, 707/4, 707/9

FIELD-OF-SEARCH: 707/3, 707/4, 707/9, 707/101, 707/102, 707/103, 707/104, 717/104, 705/54, 709/101

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5715453</u>	February 1998	Stewart	707/104
<u>5737592</u>	April 1998	Nguyen et al.	707/4
<u>5802518</u>	September 1998	Karaev et al.	707/9
<u>6003039</u>	December 1999	Barry et al.	707/103
<u>6016394</u>	January 2000	Walker	717/104
<u>6314408</u>	November 2001	Salas et al.	705/54
<u>6317751</u>	November 2001	Yeger et al.	707/104
<u>6453353</u>	September 2002	Win et al.	709/229
<u>6539396</u>	March 2003	Bowman-Amuah	707/103
<u>6681229</u>	January 2004	Cason et al.	707/101

OTHER PUBLICATIONS

Lotus Development Corp., an IBM Subsidiary. Lotus Domino: Application Development with Domino Designer, Rel. 5. USA, privately printed, 1998. i-vii. 425-430.

Lotus Development Corp., an IBM Subsidiary. Lotus Domino: Domino Enterprise Integration Guide, Rel. 5. USA, privately printed, 1998. 9-20, 107-122, 444-451.

SUN "Java II Enterprise Edition". [Copy not available as of June 15, 2002].

SUN "Java II Enterprise Edition". [Described below: AU and AV].

Sun Microsystems, Inc. Java 2 Enterprise Edition Technical Documentation . Copyright 1999. 1 page.

Sun Microsystems, Inc. Java 2 Enterprise Edition Developer's Guide . Copyright 1999. Table of Contents (7 pages) and Chapter 8 "Security" (10 pages).

ART-UNIT: 2175

PRIMARY-EXAMINER: Pardo; Thuy N.

ATTY-AGENT-FIRM: Schiesser; William E. Beckstrand; Shelley M

ABSTRACT:

A hybrid Notes/DB2 environment provides a requisition catalog on the Web. Client browsers are connected to a GWA infrastructure including a first network dispatcher and a virtual cluster of Domino.Go servers. The network dispatcher sprays out browser requests among configured .nsf servers in virtual server cluster. Communications from this virtual server cluster are, in turn, dispatched by a second network dispatcher in a Domino cluster. External objects, primarily for a GUI, are served in a .dfs and include graphic files, Java files, HTML images and net.data macros. The catalog is built from supplier provided flat files. A front end is provided for business logic and validation, as also is a relation database backend. HTML forms are populated using relational database agents. A role table is used for controlling access both to Notes code and DB2 data. Large amounts of data are quickly transferred using an intermediate agent and window.

15 Claims, 21 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KMC	Drawn Desc	Image
----------------------	-----------------------	--------------------------	-----------------------	------------------------	--------------------------------	----------------------	---------------------------	--	--	------------------------	---------------------	----------------------------	-----------------------

2. Document ID: US 6760727 B1

L4: Entry 2 of 8

File: USPT

Jul 6, 2004

US-PAT-NO: 6760727

DOCUMENT-IDENTIFIER: US 6760727 B1

TITLE: System for customer contact information management and methods for using same

DATE-ISSUED: July 6, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schroeder; Kim A.	Murray	UT		
Jenkins; Aaron	Kaysville	UT		
Donnelly; Keric P.	Salt Lake City	UT		
Smith; Greysen	SLC	UT		
Anderson; Shawn	Draper	UT		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Convergys CMG Utah, Inc.	South Jordon	UT			02

APPL-NO: 09/ 467378 [PALM]

DATE FILED: December 21, 1999

PARENT-CASE:

The present application claims priority under 35 U.S.C. .sctn.119(e) to provisional application No. 60/146,515, filed Jul. 30, 1999, the entirety of which is incorporated herein by reference.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/10; 707/1, 707/9

US-CL-CURRENT: 707/10; 707/1, 707/9

FIELD-OF-SEARCH: 707/1-10, 707/100-104.1, 707/200

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>4893301</u>	January 1990	Andrews et al.	
<u>5073890</u>	December 1991	Danielsen	
<u>5291550</u>	March 1994	Levy et al.	
<u>5291551</u>	March 1994	Conn et al.	
<u>5309513</u>	May 1994	Rose	
<u>5351235</u>	September 1994	Lahtinen	370/259
<u>5392345</u>	February 1995	Otto	
<u>5440585</u>	August 1995	Partridge, III	
<u>5448555</u>	September 1995	Bremer et al.	
<u>5452350</u>	September 1995	Reynolds et al.	

<u>5459780</u>	October 1995	Sand	
<u>5537436</u>	July 1996	Bottoms et al.	
<u>5546452</u>	August 1996	Andrews et al.	
<u>5555299</u>	September 1996	Maloney et al.	379/212.01
<u>5661718</u>	August 1997	Bremer et al.	
<u>5684870</u>	November 1997	Maloney et al.	379/212.01
<u>5696811</u>	December 1997	Maloney et al.	
<u>5737726</u>	April 1998	Cameron et al.	705/7
<u>5742675</u>	April 1998	Kilander et al.	
<u>5768360</u>	June 1998	Reynolds et al.	
<u>5778060</u>	July 1998	Otto	
<u>5796791</u>	August 1998	Polcyn	379/265.09
<u>5802163</u>	September 1998	Miloslavsky	
<u>5848143</u>	December 1998	Andrews et al.	379/265.09
<u>5862211</u>	January 1999	Roush	
<u>5878130</u>	March 1999	Andrews et al.	379/265.09
<u>5884032</u>	March 1999	Bateman et al.	709/204
<u>5901209</u>	May 1999	Tannenbaum et al.	
<u>5915003</u>	June 1999	Bremer et al.	
<u>5926816</u>	July 1999	Bauer et al.	707/8
<u>5948059</u>	September 1999	Woo et al.	709/206
<u>6055513</u>	April 2000	Katz et al.	705/26
<u>6067525</u>	May 2000	Johnson et al.	705/10
<u>6069946</u>	May 2000	Lieuwen	379/211.01
<u>6070142</u>	May 2000	McDonough et al.	705/7
<u>6072867</u>	June 2000	Lieuwen	379/220.01
<u>6115693</u>	September 2000	McDonough et al.	705/10
<u>6138009</u>	October 2000	Birgerson	455/419
<u>6185194</u>	February 2001	Musk et al.	370/260
<u>6411684</u>	June 2002	Cohn et al.	379/88.14
<u>6493447</u>	December 2002	Goss et al.	379/265.09

ART-UNIT: 2177

PRIMARY-EXAMINER: Robinson; Greta

ASSISTANT-EXAMINER: Black; Linh

ATTY-AGENT-FIRM: Frost Brown Todd LLC

ABSTRACT:

A system and method for providing complete customer contact management across a variety of customer contact channels so that businesses, or clients (i.e., those businesses using the services of the system and method of the present invention), can manage and develop relationships with their customers, to increase sales and enhance business productivity. In addition, clients can monitor customer's preferences for products or services, so that the client can develop new products and services or modify existing products and services to meet the demand of the customers. An improved system retrieves customer contact information from a plurality of contact channels, storing that information, and performing subsequent processes,

such as making the data available to the clients in the form of, for example, reports.

52 Claims, 22 Drawing figures

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Abstract](#) | [Claims](#) | [KWC](#) | [Draw Desc](#) | [Image](#)

3. Document ID: US 6681229 B1

L4: Entry 3 of 8

File: USPT

Jan 20, 2004

US-PAT-NO: 6681229

DOCUMENT-IDENTIFIER: US 6681229 B1

TITLE: System and method for providing a relational database backend

DATE-ISSUED: January 20, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Cason; Stanley P.	Johnson City	NY		
Cohen; Ira H.	Endwell	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 09/ 656037 [PALM]

DATE FILED: September 7, 2000

PARENT-CASE:

CROSS REFERENCES TO RELATED APPLICATIONS The following U.S. patent applications filed concurrently herewith are assigned to the same assignee hereof and contain subject matter related, in certain respects, to the subject matter of the present application, the teachings of which applications are incorporated herein by this reference: Serial No. 09/657,215, entitled "System and Method for Clustering Servers for Performance and Load Balancing"; Serial No. 09/657,216, entitled "System and Method for Front End Business Logic and Validation"; Serial No. 09/657,217, entitled "System and Method for Data Transfer With Respect to External Applications"; Serial No. 09/656,803, entitled "System and Method for Providing a Role Table GUI via Company Group"; Serial No. 09/656,967, entitled "System and Method for Populating HTML Forms Using Relational Database Agents"; Serial No. 09/657,196, entitled "System and Method for Catalog Administration Using Supplier Provided Flat Files"; and Serial No. 09/657,195, entitled "System and Method for Providing an Application Navigator Client Menu Side Bar".

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/101, 707/10, 709/328

US-CL-CURRENT: 707/101; 707/10, 719/328

FIELD-OF-SEARCH: 707/1-10, 707/100-103, 709/200-203, 709/328-332, 711/136, 711/137, 711/110

PRIOR-ART-DISCLOSED:

U. S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5640550</u>	June 1997	Coker	707/101
<u>5715453</u>	February 1998	Stewart	395/615
<u>5974407</u>	October 1999	Sacks	707/1
<u>6058373</u>	May 2000	Blinn et al.	705/22
<u>6343287</u>	January 2002	Kumar et al.	707/100
<u>6499036</u>	December 2002	Gurevich	707/103R

ART-UNIT: 2172

PRIMARY-EXAMINER: Corrielus; Jean M.

ASSISTANT-EXAMINER: Hwang; Joon Hwan

ATTY-AGENT-FIRM: Beckstrand; Shelley M

ABSTRACT:

A hybrid Notes/DB2 environment provides a requisition catalog on the Web. Client browsers are connected to a GWA infrastructure including a first network dispatcher and a virtual cluster of Domino.Go servers. The network dispatcher sprays out browser requests among configured .nsf servers in virtual server cluster. Communications from this virtual server cluster are, in turn, dispatched by a second network dispatcher servers in a Domino cluster. External objects, primarily for a GUI, are served in a .dfs and include graphic files, Java files, HTML images and net.data macros. The catalog is built from supplier provided flat files. A front end is provided for business logic and validation, as also is a relation database backend. HTML forms are populated using relational database agents. A role table is used for controlling access both to Notes code and DB2 data. Large amounts of data is quickly transferred using an intermediate agent and window.

14 Claims, 21 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draft Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	------------	-------

4. Document ID: US 6611851 B2

L4: Entry 4 of 8

File: USPT

Aug 26, 2003

US-PAT-NO: 6611851

DOCUMENT-IDENTIFIER: US 6611851 B2

TITLE: Storage sub-system having expanded data read

DATE-ISSUED: August 26, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ishikawa; Atsushi	Minamiashigara			JP
Matsumoto; Yoshiko	Minamiashigara			JP
Takamoto; Kenichi	Odawara			JP

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Hitachi, Ltd.	Tokyo			JP	03

APPL-NO: 10/ 090843 [PALM]

DATE FILED: March 6, 2002

PARENT-CASE:

This is a continuation application of U.S. Ser. No. 09/236,443, filed Jan. 25, 1999, now U.S. Pat. No. 6,374,269 now allowed.

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	10-012457	January 26, 1998

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/205; 707/201, 707/202

US-CL-CURRENT: 707/205; 707/201, 707/202

FIELD-OF-SEARCH: 707/205, 707/201, 707/202, 707/204, 711/112, 711/113, 714/710, 714/7, 714/10

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5455926</u>	October 1995	Keele et al.	
<u>5515500</u>	May 1996	Mizuno et al.	
<u>5734859</u>	March 1998	Yorimitsu et al.	
<u>6237046</u>	May 2001	Ohmura et al.	

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
7-200191	August 1995	JP	

ART-UNIT: 2172

PRIMARY-EXAMINER: Corrielus; Jean M.

ATTY-AGENT-FIRM: Mattingly, Stanger & Malur, P.C.

ABSTRACT:

A storage sub-system employs a staging control information table by which staging of data to be read and redundant data thereof can be executed together to reduce response time in the event of a data read failure. The staging control information table also permits pre-read staging to be executed in the forward, backward or both the forward and backward directions, to reduce response time.

1 Claims, 7 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	KWIC	Draw. Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	--------	------	------------	-------

 5. Document ID: US 6374269 B1

L4: Entry 5 of 8

File: USPT

Apr 16, 2002

US-PAT-NO: 6374269

DOCUMENT-IDENTIFIER: US 6374269 B1

TITLE: Storage sub-system having expanded data read

DATE-ISSUED: April 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ishikawa; Atsushi	Minamiashigara			JP
Matsumoto; Yoshiko	Minamiashigara			JP
Takamoto; Kenichi	Odawara			JP

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Hitachi, Ltd.	Tokyo			JP	03

APPL-NO: 09/ 236443 [PALM]

DATE FILED: January 25, 1999

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	10-012457	January 26, 1998

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/205; 707/201, 707/202, 707/204, 711/112, 711/113, 711/114, 714/710

US-CL-CURRENT: 707/205; 707/201, 707/202, 707/204, 711/112, 711/113, 711/114, 714/710

FIELD-OF-SEARCH: 714/6, 714/20, 714/7, 714/210, 714/710, 711/112, 711/113, 711/114, 711/111, 711/129, 711/137, 711/155, 711/160, 711/161, 711/202, 707/204, 707/202, 707/201, 707/205

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5455926</u>	October 1995	Keele et al.	395/404
<u>5515500</u>	May 1996	Mizuno et al.	714/7
<u>5734859</u>	March 1998	Yorimitsu et al.	395/440
<u>6237046</u>	May 2001	Ohmura et al.	710/1

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO 7-200191	PUBN-DATE August 1995	COUNTRY JP	US-CL
----------------------------	--------------------------	---------------	-------

ART-UNIT: 2172

PRIMARY-EXAMINER: Corrielus; Jean M.

ATTY-AGENT-FIRM: Mattingly, Stanger & Malur, P.C.

ABSTRACT:

A storage sub-system employs a staging control information table by which staging of data to be read and redundant data thereof can be executed together to reduce response time in the event of a data read failure. The staging control information table also permits pre-read staging to be executed in the forward, backward or both the forward and backward directions, to reduce response time.

23 Claims, 7 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	[REDACTED]	[REDACTED]	[REDACTED]	Claims	KWIC	Drawn Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	------------	------------	------------	--------	------	------------	-------

 6. Document ID: US 6212524 B1

L4: Entry 6 of 8

File: USPT

Apr 3, 2001

US-PAT-NO: 6212524

DOCUMENT-IDENTIFIER: US 6212524 B1

TITLE: Method and apparatus for creating and populating a datamart

DATE-ISSUED: April 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Weissman; Craig David	Belmont	CA		
Walsh; Gregory Vincent	Cupertino	CA		
Slater, Jr.; Lynn Randolph	Fremont	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
E.piphany, Inc.	San Mateo	CA			02

APPL-NO: 09/ 073752 [PALM]

DATE FILED: May 6, 1998

PARENT-CASE:

CROSS REFERENCES TO RELATED APPLICATIONS This application relates to the following group of applications. Each application in the group relates to, and incorporates by reference, each other application in the group. The invention of each application is assigned to the assignee

of this invention. The group of applications includes the following. U.S. patent application Ser. No. 09/385,119, entitled "Method and Apparatus for Creating a Well-Formed Database System Using a Computer," filed Aug. 27, 1999, and having inventors Craig David Weissman, Greg Vincent Walsh and Eliot Leonard Wegbreit. U.S. patent application Ser. No. 09/073,752, entitled "Method and Apparatus for Creating and Populating a Datamart," filed May 6, 1998, and having inventors Craig David Weissman, Greg Vincent Walsh and Lynn Randolph Slater, Jr. U.S. patent application Ser. No. 09/073,733, entitled "Method and Apparatus for Creating Aggregates for Use in a Datamart," filed May 6, 1998, and having inventors Alton Rauer, Gregory Vincent Walsh, John P. McCaskey, Craig David Weissman and Jeremy A. Rassen. U.S. patent application Ser. No. 09/073,753, entitled "Method and Apparatus for Creating a Datamart and for Creating a Query Structure for the Datamart," filed May 6, 1998, and having inventors Jeremy A. Rassen, Emile Litvak, abhi a. shelat, John P. McCaskey and Alton Rauer.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/101; 707/3

US-CL-CURRENT: 707/101; 707/3

FIELD-OF-SEARCH: 707/1-10, 707/100-104, 707/200-206

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5386556</u>	January 1995	Hedin et al.	707/4
<u>5550971</u>	August 1996	Brunner et al.	707/3
<u>5659724</u>	August 1997	Borgida et al.	707/3
<u>5675785</u>	October 1997	Hall et al.	707/102
<u>5806060</u>	September 1998	Borgida et al.	707/3
<u>5995958</u>	November 1999	Xu	707/3

OTHER PUBLICATIONS

- Kimball, R., "The Data Warehouse Toolkit", (1996) John-Wiley & Sons, Inc., 388 pages (includes CD ROM).
- Chawathe, S. et al., "Change Detection in Hierarchically Structured Information", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 493-504.
- Chawathe, S. et al., "Meaningful Change Detection in Structured Data", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 26-37.
- Labio, W. et al. "Efficient Snapshot Differential Algorithms for Data Warehousing", Department of Computer Science, Stanford University, (1996), pp. 1-13.
- Wiener, J. et al., "A System Prototype for Warehouse View Maintenance", The Workshop on Materialized Views, pp. 26-33, Montreal, Canada, Jun. 1996.
- Kawaguchi, A. et al., "Concurrency Control Theory for Deferred Materialized Views", Database Theory-ICDT '97, Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 306-320.
- Zhuge, Y. et al., "Consistency Algorithms for Multi-Source Warehouse View Maintenance", Distributed and Parallel Databases, vol. 6, pp. 7-40 (1998), Kluwer Academic Publishers.
- Zhuge, Y. et al., "View Maintenance in a Warehousing Environment", SIGMOD Record, vol. 24, No. 2, Jun. 1995, pp. 316-327.
- Widom, J., "Research Problems in Data Warehousing", Proc. of 4th Int'l Conference on Information and Knowledge Management (CIKM), Nov. 1995, 6 pages.
- Yang, J. et al., "Maintaining Temporal Views Over Non-Historical Information Sources For Data Warehousing", Advances in Database Technology--EDBT '98 , Proceedings of the 6th International

Conference on Extending Database Technology, Valencia, Spain, Mar. 1998, pp. 389-403.
Quass, D., "Maintenance Expressions for Views with Aggregation", Proceedings of the 21st International Conference on Very Large Data Bases, IEEE, Zurich, Switzerland, (Sep. 1995), 9 pages.
Mumick, I. et al., "Maintenance of Data Cubes and Summary Tables in a Warehouse", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 100-111.
Huyn, N., "Multiple-View Self-Maintenance in Data Warehousing Environments", Proceedings of the 23rd International Conference on Very Large Data Bases, IEEE, (1997), pp. 26-35.
Quass, D. et al., "Making Views Self-Maintainable for Data Warehousing", Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems, IEEE, Dec. 1996, pp. 158-169.
Quass, D. et al., "On-Line Warehouse View Maintenance", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 393-404.
Gupta, H., "Selection of Views to Materialize in a Data Warehouse", Database Theory--ICDT '97 , Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 98-112.
Harinarayan, V. et al., "Implementing Data Cubes Efficiently", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 205-216.
Gupta, H. et al., "Index Selection for OLAP", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 208-219.
Labio, W. et al., "Physical Database Design for Data Warehouses", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 277-288.
Gupta, A. et al., "Aggregate-Query Processing in Data Warehousing Environments", Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995, 358-369.
O'Neill, P. et al., "Improved Query Performance with Variant Indexes", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 38-49.
McAlpine, G. et al., "Integrated Information Retrieval in a Knowledge Worker Support System", Proc. of the Intl. Conf. on Research and Development In Information Retrieval (SIGIR), Cambridge, MA, Jun. 25-28, 1989, Conf. 12, pp. 48-57.
Tsuda, K. et al., "IconicBrowser: An Iconic Retrieval System for Object-Oriented Databases", Proc. of the IEEE Workshop on Visual Languages, Oct. 4, 1989, pp. 130-137.
"Multiple Selection List Presentation Aids Complex Search", IBM Technical Disclosure Bulletin, vol. 36, No. 10, Oct. 1993, pp. 317-318.

ART-UNIT: 271

PRIMARY-EXAMINER: Ho; Ruay Lian

ATTY-AGENT-FIRM: Skjerven Morrill MacPherson LLP Marino; Fabio E.

ABSTRACT:

A method of generating a datamart is described. The datamart includes tables having rows and columns. The method comprises accessing a description of a schema. The schema defines the relationships between the tables and columns. The description further defines how data is to be manipulated and used to populate the tables in the datamart. That is, the description defines the semantic meaning of the data. The description is further used to create a set of commands to create the tables. The commands are executed causing the creation of the tables. Importantly, when the semantic meaning is associated with the column and rows, programs for manipulating and propagating data into those columns and rows are automatically defined. Previously, consultants would have to hand code the creation, manipulation, and population programs for a datamart. Thus, the amount of work required to create and populate the datamart is significantly reduced.

21 Claims, 48 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Image	Image	Claims	KWC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-------	-------	--------	-----	-----------	-------

7. Document ID: US 6189004 B1

L4: Entry 7 of 8

File: USPT

Feb 13, 2001

US-PAT-NO: 6189004

DOCUMENT-IDENTIFIER: US 6189004 B1

TITLE: Method and apparatus for creating a datamart and for creating a query structure for the datamart

DATE-ISSUED: February 13, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rassen; Jeremy A.	Sunnyvale	CA		
Litvak; Emile	Mountain View	CA		
shelat; abhi a.	Mountain View	CA		
McCaskey; John P.	Mountain View	CA		
Rauer; Allon	Mountain View	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
E. Piphany, Inc.	San Mateo	CA			02

APPL-NO: 09/ 073753 [PALM]

DATE FILED: May 6, 1998

PARENT-CASE:

CROSS REFERENCES TO RELATED APPLICATIONS This application relates to the following group of applications. Each application in the group relates to, and incorporates by reference, each other application in the group. The invention of each application is assigned to the assignee of this invention. The group of applications includes the following. U.S. patent application Ser. No. 09/073,748, entitled "Method and Apparatus for Creating a Well-Formed Database System Using a Computer," filed May 6, 1998, and having inventors Craig David Weissman, Greg Vincent Walsh, and Eliot Leonard Wegbreit. U.S. patent application Ser. No. 09/073,752, entitled "Method and Apparatus for Creating and Populating a Datamart," filed May 6, 1998, and having inventors Craig David Weissman, Greg Vincent Walsh and Lynn Randolph Slater, Jr. U.S. patent application Ser. No. 09/073,733, entitled "Method and Apparatus for Creating Aggregates for Use in a Datamart," filed May 6, 1998, and having inventors Allon Rauer, Gregory Vincent Walsh, John P. McCaskey, Craig David Weissman and Jeremy A. Rassen. U.S. patent application Ser. No. 09/073,753, entitled "Method and Apparatus for Creating a Datamart and for Creating a Query Structure for the Datamart," filed May 6, 1998, and having inventors Jeremy A. Rassen, Emile Litvak, abhi a. shelat, John P. McCaskey and Allon Rauer.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/3; 707/4, 707/102

US-CL-CURRENT: 707/3; 707/102, 707/4

FIELD-OF-SEARCH: 707/1-10, 707/100-104, 707/200-206

PRIOR-ART-DISCLOSED:

U. S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5386556</u>	January 1995	Hedin et al.	707/4
<u>5550971</u>	August 1996	Brunner et al.	707/3
<u>5659724</u>	August 1997	Borgida et al.	707/3
<u>5675785</u>	October 1997	Hall et al.	707/102
<u>5806060</u>	September 1998	Borgida et al.	707/3
<u>5995958</u>	November 1999	Xu	707/3

OTHER PUBLICATIONS

- Kimball, R., "The Data Warehouse Toolkit", (1996) John-Wiley & Sons, Inc., 388 pages (includes CD ROM).
- Chawathe, S. et al., "Change Detection in Hierarchically Structured Information", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 493-504.
- Chawathe, S. et al., "Meaningful Change Detection in Structured Data", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 26-37.
- Labio, W. et al., "Efficient Snapshot Differential Algorithms for Data Warehousing", Department of Computer Science, Stanford University, (1996), pp. 1-13.
- Wiener, J. et al., "A System Prototype for Warehouse View Maintenance", The Workshop on Materialized Views, pp. 26-33, Montreal, Canada, Jun. 1996.
- Kawaguchi, A. et al., "Concurrency Control Theory for Deferred Materialized Views", Database Theory--ICDT '97, Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 306-320.
- Zhuge, Y. et al., "Consistency Algorithms for Multi-Source Warehouse View Maintenance", Distributed and Parallel Databases, vol. 6, pp. 7-40 (1998), Kluwer Academic Publishers.
- Zhuge, Y. et al., "View Maintenance in a Warehousing Environment", SIGMOD Record, vol. 24, No. 2, Jun. 1995, pp. 316-327.
- Widom, J., "Research Problems in Data Warehousing", Proc. of 4th Int'l Conference on Information and Knowledge Management (CIKM), Nov. 1995, 6 pages.
- Yang, J. et al., "Maintaining Temporal Views Over Non-Historical Information Sources For Data Warehousing", Advances in Database Technology--EDBT '98, Proceedings of the 6th International Conference on Extending Database Technology, Valencia, Spain, Mar. 1998, pp. 389-403.
- Quass, D., "Maintenance Expressions for Views with Aggregation", Proceedings of the 21st International Conference on Very Large Data Bases, IEEE, Zurich, Switzerland, (Sep. 1995), 9 pages.
- Mumick, I. et al., "Maintenance of Data Cubes and Summary Tables in a Warehouse", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 100-111.
- Huyn, N., "Multiple-View Self-Maintenance in Data Warehousing Environments", Proceedings of the 23rd International Conference on Very Large Data Bases, IEEE, (1997), pp. 26-35.
- Quass, D. et al., "Making Views Self-Maintainable for Data Warehousing", Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems, IEEE, Dec. 1996, pp. 158-169.
- Quass, D. et al., "On-Line Warehouse View Maintenance", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 393-404.
- Gupta, H., "Selection of Views to Materialize in a Data Warehouse", Database Theory--ICDT '97, Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 98-112.
- Harinarayan, V. et al., "Implementing Data Cubes Efficiently", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 205-216.
- Gupta, H. et al., "Index Selection for OLAP", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 208-219.
- Labio, W. et al., "Physical Database Design for Data Warehouses", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 277-288.
- Gupta, A. et al., "Aggregate-Query Processing in Data Warehousing Environments", Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995, pp. 358-369.
- O'Neill, P. et al., "Improved Query Performance with Variant Indexes", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 38-49.

McAlpine, G. et al., "Integrated Information Retrieval in a Knowledge Worker Support System", Proc. of the Intl. Conf. on Research and Development In Information Retrieval (SIGIR), Cambridge, MA, Jun. 25-28, 1989, Conf. 12, pp. 48-57.

Tsuda, K. et al., "IconicBrowser: An Iconic Retrieval System for Object-Oriented Databases", Proc. of the IEEE Workshop on Visual Languages, Oct. 4, 1989, pp. 130-137.

"Multiple Selection List Presentation Aids Complex Search", IBM Technical Disclosure Bulletin, vol. 36, No. 10, Oct. 1993, pp. 317-318.

ART-UNIT: 271

PRIMARY-EXAMINER: Ho; Ruay Lian

ATTY-AGENT-FIRM: Wilson Sonsoni Goodrich & Rosati

ABSTRACT:

A method for automatically defining a query interface for a datamart is described. The datamart includes fact and dimension tables. The method comprises accessing a schema description and a query interface description for the datamart. The schema description specifies a schema, which in turn, defines the relationships between the fact tables and dimension tables of the datamart. The query interface description specifies the fields, related to the schema description, that can be used in a query and the way in which results are to be presented to the user. The fields correspond to columns and rows in the fact tables. The schema description is used to create a first set of commands to create and populate the fact and dimension tables. Additionally, a second set of commands to create the query interface is created. Some commands of the first set of commands are executed causing the creation and population of the tables. Some commands of the second set of commands are executed causing the creation of a user interface. A query is generated using the user interface. The query is sent to the system for processing. The results of the query are presented to the user according the second set of commands.

9 Claims, 43 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	-----------	-------

8. Document ID: US 6161103 A

L4: Entry 8 of 8

File: USPT

Dec 12, 2000

US-PAT-NO: 6161103

DOCUMENT-IDENTIFIER: US 6161103 A

TITLE: Method and apparatus for creating aggregates for use in a datamart

DATE-ISSUED: December 12, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rauer; Allon	Mountain View	CA		
Walsh; Gregory Vincent	Cupertino	CA		
McCaskey; John P.	Mountain View	CA		
Weissman; Craig David	Belmont	CA		
Rassen; Jeremy A.	Sunnyvale	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Epiphany, Inc.	San Mateo	CA			02

APPL-NO: 09/ 073733 [PALM]

DATE FILED: May 6, 1998

PARENT-CASE:

CROSS REFERENCES TO RELATED APPLICATIONS This application relates to the following group of applications. Each application in the group relates to, and incorporates by reference, each other application in the group. The invention of each application is assigned to the assignee of this invention. The group of applications includes the following. U.S. patent application Ser. No. 09/073,748, entitled "Method and Apparatus for Creating a Well-Formed Database System Using a Computer," filed May 6, 1998, and having inventors Craig David Weissman, Greg Vincent Walsh and Eliot Leonard Wegbreit. U.S. patent application Ser. No. 09/073,752, entitled "Method and Apparatus for Creating and Populating a Datamart," filed May 6, 1998, and having inventors Craig David Weissman, Greg Vincent Walsh and Lynn Randolph Slater, Jr. U.S. patent application Ser. No. 09/073,733, entitled "Method and Apparatus for Creating Aggregates for Use in a Datamart," filed May 6, 1998, and having inventors Alton Rauer, Gregory Vincent Walsh, John P. McCaskey, Craig David Weissman and Jeremy A. Rassen. U.S. patent application Ser. No. 09/073,753, entitled "Method and Apparatus for Creating a Datamart and for Creating a Query Structure for the Datamart," filed May 6, 1998, and having inventors Jeremy A. Rassen, Emile Litvak, abhi a. shelat, John P. McCaskey and Alton Rauer.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/4; 707/1, 707/3

US-CL-CURRENT: 707/4; 707/1, 707/3

FIELD-OF-SEARCH: 707/1-10, 707/200-208, 707/100-104

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5386556	January 1995	Hedin et al.	707/4
5550971	August 1996	Brunner et al.	707/3
5659724	August 1997	Borgida et al.	707/3
5675785	October 1997	Hall et al.	707/102
5806060	September 1998	Borgida et al.	707/3
5995958	November 1999	Xu	707/3

OTHER PUBLICATIONS

McAlpine, G. et al., "Integrated Information Retrieval in a Knowledge Worker Support System", Proc. of the Intl. Conf. on Research and Development in Information Retrieval (SIGIR), Cambridge, MA, Jun. 25-28, 1989, Conf. 12, pp. 48-57.

Tsuda, K. et al., "IconicBrowser: An Iconic Retrieval System for Object-Oriented Databases", Proc. of the IEEE Workshop on Visual Languages, Oct. 4, 1989, pp. 130-137.

"Multiple Selection List Presentation Aids Complex Search", IBM Technical Disclosure Bulletin, vol. 36, No. 10, Oct. 1993, pp. 317-318.

Kimball, R., "The Data Warehouse Toolkit", (1996) John-Wiley & Sons, Inc., 388 pages (includes CD ROM).

Chawathe, S. et al., "Change Detection in Hierarchically Structured Information", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 493-504.

Chawathe, S. et al., "Meaningful Change Detection in Structured Data", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 26-37.

Labio, W. et al., "Efficient Snapshot Differential Algorithms for Data Warehousing", Department of Computer Science, Stanford University, (1996), pp. 1-13.

Wiener, J. et al., "A System Prototype for Warehouse View Maintenance", The Workshop on Materialized Views, pp. 26-33, Montreal, Canada, Jun. 1996.

Kawaguchi, A. et al., "Concurrency Control Theory for Deferred Materialized Views", Database Theory-ICDT '97, Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 306-320.

Zhuge, Y. et al., "Consistency Algorithms for Multi-Source Warehouse View Maintenance", Distributed and Parallel Databases, vol. 6, pp. 7-40 (1998), Kluwer Academic Publishers.

Zhuge, Y. et al., "View Maintenance in a Warehousing Environment", SIGMOD Record, vol. 24, No. 2, Jun. 1995, pp. 316-327.

Wisdom, J. "Research Problems in Data Warehousing", Proc. of 4th Int'l Conference on Information and Knowledge Management (CIKM), Nov. 1995, 6 pages.

Yang, J. et al., "Maintaining Temporal Views Over Non-Historical Information Sources For Data Warehousing", Advances in Database Technology--EDBT '98, Proceedings of the 6th International Conference on Extending Database Technology, Valencia, Spain, Mar. 1998, pp. 389-403.

Quass, D., "Maintenance Expressions for Views with Aggregation", Proceedings of the 21st International Conference on Very Large Data Bases, IEEE, Zurich, Switzerland, (Sep. 1995), 9 pages.

Mumick, I. et al., "Maintenance of Data Cubes and Summary Tables in a Warehouse", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 100-111.

Huyn, N., "Multiple-View Self-Maintenance in Data Warehousing Environments", Proceedings of the 23rd International Conference on Very Large Data Bases, IEEE, (1997), pp. 26-35.

Quass, D. et al., "Making Views Self-Maintainable for Data Warehousing", Proceedings of the Fourth International Conference, on Parallel and Distributes Information Systems, IEEE, Dec. 1996, pp. 158-169.

Gupta, H. "Selection of Views to Materialize in a Data Warehouse", Database Theory--ICDT '97, Proceedings of the 6th International Conference, Delphi, Greece, Jan. 1997, pp. 98-112.

Harinarayan, V. et al., "Implementing Data Cubes Efficiently", SIGMOD Record, vol. 25, No. 2, Jun. 1996, pp. 205-216.

Gupta, H. et al., "Index Selection for OLAP", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 208-219.

Labio, W. et al., "Physical Database Design for Data Warehouses", IEEE Paper No. 1063-6382/97, IEEE (1997), pp. 277-288.

Gupta, A. et al., "Aggregate-Query Processing in Data Warehousing Environments", Proceedings of the 21st VLDB Conference, Zurich, Switzerland, Sep. 1995, pp. 358-369.

O'Neill, P. et al., "Improved Query Performance with Variant Indexes", Proceedings of the 1997 ACM SIGMOD International Conference, ACM Press, 1997, pp. 38-49.

ART-UNIT: 271

PRIMARY-EXAMINER: Ho; Ruay Lian

ATTY-AGENT-FIRM: Wilson, Sonsini, Goodrich & Rosati

ABSTRACT:

A method for automatically defining aggregates for use in a datamart is described. The datamart includes fact and dimension tables. The method comprises accessing a schema description and an aggregates description for the datamart. The schema description specifies a schema, which in turn, defines the relationships between the fact tables and dimension tables of the datamart. The aggregates description specifies the aggregates, which define, from the schema definition, which aggregate tables are to be created from the fact tables and dimension tables in the datamart. The data in the aggregates correspond to the pre-computed results of specific types of queries. In response to a query, the aggregates can be searched to determine an appropriate

aggregate to use in response to that query. The schema description is used to create a first set of commands to create and populate the fact and dimension tables. Additionally, a second set of commands to create, populate and access, the aggregates are also created from the aggregates description. Some of the commands of the first set of commands are executed causing the creation and population of the tables. Some of the commands of the second set of commands are executed causing the creation of the aggregate tables. Some of the remaining commands of the second set of commands are executed to populate the aggregate tables from the populated fact and dimension tables.

11 Claims, 43 Drawing figures

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Claims](#) | [KMM](#) | [Draw Desc](#) | [Image](#)

[Clear](#) | [Generate Collection](#) | [Print](#) | [Fwd Refs](#) | [Bkwd Refs](#) | [Generate OACS](#)

Term	Documents
READ	524572
READS	124613
(3 AND READ).USPT.	8
(L3 AND READ).USPT.	8

Display Format: [FRO](#) | [Change Format](#)

[Previous Page](#) [Next Page](#) [Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#) [Next Doc](#) [Go to Doc#](#) [Generate Collection](#) [Print](#)

L4: Entry 1 of 8

File: USPT

Aug 3, 2004

DOCUMENT-IDENTIFIER: US 6772167 B1

TITLE: System and method for providing a role table GUI via company group

Brief Summary Text (10):

In a hybrid requisition catalog system for use in a web environment, much of the application data is stored in, for example, IBM DB2 tables. However, a web interface written in Java script and in HTML does not have functionality for connecting to DB2 and reading data from DB2 tables. No function is provided in Java script and in HTML to connect to or access such a database. It is not presently possible to make the connection by connecting to the database and executing SQL queries. Consequently, there is a need in the art for a system and method utilizing an existing infrastructure including Lotus Notes, Domino Go, and DB2 to combine HTML and Java script web presentation with DB2 data.

Brief Summary Text (11):

In a requisition catalog system for use in a web environment by a large enterprise, there must be provide a way to deal with web sites that exist outside of a firewall, or internal applications within the firewall but outside of the requisition catalog (Req/Cat Web or RCW) application. That is, a system and method is required for transferring a large quantity of data back from such a web site or application to the RCW application in a timely manner. One possible way is to send data on the universal resource locator (URL). However, such a URL is limited to 1K bytes, which is not enough to do the task quickly for the quantities of information required. Information can be put into a frame, and that information can be read, but only if the information is written and read by the same server.

Brief Summary Text (13):

Consequently, in the new environment (with spoofing inhibited) a problem is presented: if a RCW application needs to access an outside supplier site for information to get back through one its own the frames, as soon as data is written by that outside supplier site into one of the frames owned by RCW, ownership of the frame is transferred from RCW to the supplier; RCW no longer owns the frame and cannot access the information. A system and method is needed to enable transfer of data from a supplier site to a RCW application which does not entail frame spoofing.

Detailed Description Text (23):

By using frames, a large majority of preprocessing can be performed dynamically, on the client, reducing the number of trips back and forth to the server. This is a tremendous boost to performance. The web screen described hereafter is not he result of a Notes form, but rather of a dynamically generated HTML/javascript code produced by a displayReqHeader() function. This function dynamically writes html and javascript code into the content frame of the application. The javascript function is coded in a displayreq.js file stored on the filesystem and loaded into a jsCode frame by a source (<script src=".//js/displayreq.js">) command in a jscode.html file at the time when the initial frameset is loading. A displayReqHeader() function is called from several places in the application to redisplay the requisition information in the content frame. This screen is called any time a WebReq Lotus Notes form is opened by an OpenForm command for a new requisition, or by an OpenDocument command when an existing document is opened. OpenForm occurs when the displayReqHeader() javascript function is called as the last part of an OnLoad event coded in the HTML-Attributes property of the WebReq form. Any time an existing document is opened that was saved with Form-WebReq, the OnLoad event causes the displayReqHeader() javascript to be run to OpenDocument. Any time a content frame has been loaded with some other page during the processing of a requisition, and the user performs an

action to return to the requisition in process, the dispLyReqHeader() javascript function is called directly. This form reads the information stored in reData frame and dynamically fills the content frame with this screen.

Detailed Description Text (24):

Referring to FIGS. 4 and 5, as will be more fully described hereafter, a screen display includes header frame 470, navigation frame 472, footer frame 474, temporary data frame 476, request data frame 478, and content frame 480. The tempData frame 476 is used as a temporary holder for information, and to direct calls dynamically, while keeping the current data in the screen, and making the return data available to the application.

Detailed Description Text (43):

On the other side, instead of editing an existing document as is done with Notes, Req/Cat Web executes createnewdocument. As the document opens, Lotus Notes gives the query webqueryopenagent, and this is also written in Lotus Script, which has access to data base api's 202, where data from many database 210 tables may be read to construct a Lotus document from DB2.

Detailed Description Text (49):

DB2Base 224 is extended by the DAPI 208 programmer for each DAPI 202 instance that is needed. It contains methods 240, 241 for connecting to and disconnecting from the database, a method 242 for defining the number of rows to return at a time, a method 243 for getting the next group of records, methods for reads 244, inserts 245, updates 246, and deletes 247, commit 248 and rollback 249 options, and a flag 250 to determine if all data has been retrieved.

Detailed Description Text (63):

An application 204 like the human resources (HR) application would then need to read HR data and insert it into the employee table if the employee did not exist, or update it if something changed, or delete it if the employee no longer exists. This application developer would then only have to know the methods and properties of the userProfile class 226 in order to write the application. An example of such an application is set forth in Table 2, with reference to the steps of FIG. 14. This table sets forth the HR load routine, a batch program to read HR data from a flat file and insert it into the DB2 user profile table.

Detailed Description Text (75):

A buyer at terminal 400 accesses the staging table 392 on the web 396. He views catalog items and enters transactions with action button which transfers information from staging table 392 to production table 394. Production table 394 is referenced by req cat web 388, and staging table 392 is referenced by the catalog administration function 386 operated by the buyer 400. Typically, a buyer is member of procurement organization with responsibility for negotiating deals with suppliers. A requester 402 accesses production table 394 over web 398 to create and submit a requisition to SAP 382.

Detailed Description Text (79):

Referring to FIG. 17, a preferred embodiment of these processes are presented. In supplier system 300, supplier source data 310 is extracted and reformatted in step 312 to create catalog flat file 314 in the format specified by the enterprise. In step 316 that flat file is transmitted to the enterprise 302, as is represented by line 305, where it is accepted in step 320 into the enterprise EDI mailbox 322. In step 324, the data in the flat file in mailbox 322 is reformatted and put into generation data group (GDG) 328, a location for saving more than one file, so as to retain the last N iterations, and a archive entry made to processing log 326. In step 330, a delivery component executes to send data from GDG 328 to application server 114, as is represented by line 303, in the form of catalog flat file 340. In step 342, a delivery component receives the flat file and, as is represented by line 347, starts job scripts including MASSLOAD for reading the flat file and loading staging table 392, and as represented by line 345 alerts the buyer 352. As is represented by lines 311, 313 and 315, respectively, MASSLOAD 344 accesses database server 306 procedures catalog_s 360, product_s 362, and Req/Cat Web 364, and makes an archival entry to processing log 346.

Detailed Description Text (84):

In operation, validation procedure 364 validates the format and identifies catalog changes to product_s 362, logging those changes in file 332. It then checks a flag in catalog 366, and if the flag is on invokes procedure 350 provided catalog_s 360 does not indicate any critical errors. Validate and load procedure 350 then moves the contents of product_s staging table 362 into the appropriate production table 368, writing any errors to processing log 348. (In the event that procedure 364 does not call procedure 350, then buyer 352 intervention is required via GUI 370.) After procedure 364 completes execution, it may either stop, or if catalog 366 has a flag set on and catalog_s staging table 360 indicates no critical errors, then procedure 364 will invoke validation and migration procedure 350. After validate procedure 364 completes, it has written to prod_message_s file 332, and the buyer may use GUI 370 to read messages from file 332 and make any desired changes to staging table 362. They buyer may also choose to reject the catalog and, via step 354, contact the supplier to restart the process. This occurs if there is an error in the unit prices, which is an example of information in the catalog which a buyer is not authorized to change on his own.

Detailed Description Text (98):

In accordance with a preferred embodiment of the invention, in a Notes/DB2 hybrid environment, a Notes agent reads data from a DB2 table, and then dynamically populates that data to an HTML page. In this manner, an the existing infrastructure (including Notes, Domino Go, and DB2) is used to combine HTML and Java script web presentation with DB2 data.

Detailed Description Text (99):

Referring to FIG. 20, Notes agents 440 are used as intermediaries. Each such agent 440 reads DB2 tables 390, collects data using SQL select statements, and builds web page 442 dynamically, writing out the Java script and HTML to present the page on a Web browser, such as Web browser 100. The results of the DB2 searches also helps to determine which HTML needs to be written, something which standard HTML cannot handle. Thus, conditional logic may be used.

Detailed Description Text (101):

For example, to display a list of supplier profiles, two DB2 tables 390 must be read: one provides a list of suppliers and the other is role table 420 (FIG. 19). When role table 420 is read, the code tests the users ability to edit (country admin for country of supplier), and may display the web page differently depending thereon. A dynamic feature of the invention is that straight text may be displayed, or with text with hyperlinks to open a supplier profile, as an example.

Detailed Description Text (102):

To make and use this preferred embodiment of the invention, the following is done: 1. Use the Lotus Script lsx:lc connector connect to DB2. 2. Depending on the DB2 table being read and the functions required, write functions to Create, Read, Update, and Delete with respect to the DB2 table. 3. Write the HTML to display the page, and then have the Notes agent 440 Print these HTML commands to the browser so that they appear in a meaningful presentation to the end user. 4. Use conditional logic to change the look of the page 442 based on the results of the DB2 390 lookups.

Detailed Description Text (103):

Inputs to the method of this embodiment of the invention include the DB2 table to be read or updated, and the output includes HTML conditionally generated based on results of the DB2 table reads.

Detailed Description Text (109):

In accordance with the preferred embodiment of the invention, the supplier 300 opens a new window through normal Java script commands and writes its data into it, along with the name of the agent to run in RCW application server 114. That new window then calls one of the RCW processes, which is able to see the data because it is not in a frame and is therefore available for RCW to access. That process causes an agent to runs (in RCW) that can see that data and write it to a frame on the RCW side, resulting in RCW owning both the data and the frame. This allows RCW to reach NOTES information, and write that information to other frames. RCW owns the data and the frame. Ownership is established in the RCW application.

Detailed Description Text (112):

Referring to FIG. 21, a system and method are provided for data transfer from a externally owned site to an application owned frame set which operates as follows. The RCW application opens a supplier site URL in a temporary frame 456. Upon user selection of go to supplier 451, as is represented by step 458, frame 456 opens the supplier window 460 as a separate browser session. Two windows are now open: the original application 450 with its window underlying, and a supplier window 460 over it with the supplier URL. The reason for doing this is that the supplier requires that the browser be full frame, not in a small frame set. The primary RCW application in window 450 is quiesced to a wait state. As is represented by steps 462, the user can now select from window 460 items to buy, search, or whatever the supplier deems is appropriate for a user to order his data. The user then issues the command to submit the order. In step 444, the supplier site then gathers content from order data entered at window 460, in step 446 formats the page, and in step 448 issues a call to Req/Cat Web to open third window 464 with first agent 480. The supplier uses an enterprise specified agent name for first agent 480 when opening third window 464. Third window 464 is a window, but not a frame, and thus the Req/Cat Web can get access to it even though it is opened by the supplier. First agent 480 includes an html form command 488 which defines the processing to be done on the contents 466 of the form now displayed in third window 464, and the supplier site writes into this third window 464 unique order identifying information. Once written, the browser activates the form. Once activated, it is a program in its own right, the html 488 that was written and any java script in it will execute. One of first things it does is look at the action in the form command and determine that this is the program that will run to deal with the contents of this form. That action program, or first agent 480, is a RCW action program on the RCW server that can see contents of third window 464 because it is not in a frame, and thus ownership is not critical. In step 482, first RCW agent 480 executes a program or process that writes the contents 466 of window 464 back into its frame set (temporary frame) 456, and then calls second agent 484 which references Java script code 454 and, as is represented by step 486, access Notes data on the Notes server, add content to the requisition, and issue the commands that send the order to be stored in the requisition. This is key, Req/Cat Web has used its own process 480 to write into its own window 450, and knows who owns the data. First agent 480 process opened window 464 and writes the data to temporary frame 456 and then kicks off another process, second agent 484, that can read that data, can read and write to all of the frames 452 in the application, can access information from Lotus notes, and write all the information into the requisition.

Detailed Description Text (119):

Referring to FIG. 4, navigation frame or menu bar appears, typically, on the left of a display window, to display a plurality of menu boxes, including headers 491-494 and items 495-500. As a cursor is moved over the headers, each individual header is highlighted or some symbol 481, 483, 485, 487, respectively, rotated by, say, 45 degrees, so as to point either down or to the right, to indicate to the user the header which will, upon being clicked, toggled to either a collapsed or expanded state from its current expanded or collapsed state, respectively. As illustrated, headers 481 and 483 are in a collapsed state, and headers 485 and 487 are in an expanded state. When expanded, header 485 is expanded to show items 495-498, and header 487 to show items 499-500. The user may move the cursor to one of items 495-500, and select the item to update the data displayed in content frame 480.

Detailed Description Text (120):

Heretofore, when the user selects (clicks on) a menu header 491, there is generally a pause as the request is made to the server to obtain a new page including an expanded menu bar 472 including a display of the included items. In accordance with a preferred embodiment of the invention, when server loads the window to a client, all of the information required to load the menu bar 472 is provided, including information for the expanded menu items (but not the content frame associated with them). If a menu header 485 is clicked when in the collapsed state, the menu items 496-498 are shown or made visible and the following menu headers and items are moved relative to the expanded menu bar. If a menu item 496 is clicked, then communication with the server is required to load the content frame 480.

Detailed Description Text (122):

This is done using dynamic html (dhtml), which allows creation of divisions within a document.

These divisions are equivalent to tab items, menu items 495-500, or headers 491-494. In Netscape, these divisions are called layers. In Explorer, the layers are called divisions. These divisions can be hidden or shown, and moved relative distances on the screen. The current state of the art is to use these divisions for moving or flashing graphics, but not for business applications such as menu bars. In accordance with the present invention dhtml at the client creates subtle changes, business like, in the menu bar 472 without requiring server communications. The use of dhtml is described at developer.netscape.com.

Detailed Description Text (134):

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a computer program product or program element, or a program storage or memory device such as a solid or fluid transmission medium, magnetic or optical wire, tape or disc, or the like, for storing signals readable by a machine, for controlling the operation of a computer according to the method of the invention and/or to structure its components in accordance with the system of the invention.

Detailed Description Paragraph Table (2):

```
TABLE 2 EXAMPLE APPLICATION `in the following step, instantiate a DB2 config object, and call it db2; the database name 230, etc., is determined by instantiating the db2 config object, as defined by the core programmers` 272: Dim db2 As New DB2Config(session) `the database information is known, and can be passed to the employee profile.` 274: Print "The target DB2 database alias is "& db2.getDB2DatabaseName ()` All that must be done is to pass the DB2 class to the userProfile.` 276: Dim eps As New userProfile(db2) `Delete everything from the employee table to start the bridge.` 278: I headerDivision = "" Then Call eps.clearTable() Else Call eps.deleteEmployeesWhere("COGRP_CD= "" &headerDivision &"") End If 280: For count = 2 To records-1 `Read the next record and make sure that it can be loaded without problems` If ReadInputFile(inputFileNum, count, userid, al, cl eps, cci) Then Call eps.insertEmployee (DB2ColumnNames, BuildDB2ColumnValues()) db2kAdditions = db2Additions+1 End If Next...
```

Detailed Description Paragraph Table (3):

```
TABLE 3 WEQUERYSAVEAGENT EXAMPLE Dimension db2 As New DB2Config(session) Print "The target DB2 database alias is "& db2.getDB2DatabaseName ()` Dimension eps As New userProfile(db2) execute process 260 to selectemployee by employee id if employee does not exist, then execute process 264 to insert employee else if employee changed, then execute process 265 to update employee else (employee not changed) information to user "employee not changed" no save endif
```

Detailed Description Paragraph Table (5):

```
TABLE 5 EXAMPLE AGENT 440 This agent is invoked from the administration tab for `supplier`. It is presenting the user with a view of suppliers that have been configured on the system. Based on the access authority of the user, this list will be presented either as just text, for the general user, or as hotlinks for an administrator to then open an individual supplier profile and update it. Sub Initialize //setting up variables Dim src As New LCConnection ("db2") Dim fldLst As New LCFIELDList(100) Dim suppname As LCFIELD Dim suppcode As LCFIELD Dim InfoView As AllInfoView Dim session As New NotesSession Dim doc As NotesDocument Dim sqlQueryString As String Dim flag As Variant Dim admin As Integer Admin=True Dim lclsxSession As LCSession Dim supplierdb As notesdatabase Dim configview As notesview Dim configdoc As notesdocument Dim lookuptype As String On Error Goto errHandler Set supplierdb=session.currentdatabase Set configview = supplierdb.getview("APPVIEW") Set configdoc = configview.getFirstDocument lookuptype = configdoc.HRFormat(0) Set doc=session.DocumentContext Set InfoView=New AllInfoView //connect to DB2 InfoView.ConnectToDB2 //Query the role table for users roles and authority levels If Not InfoView.CheckAdminPrivilegesOK (doc.CurrentUserName (0))Then Admin = False End If InfoView.DisconnectDB2 If (Admin = False) And (lookuptype = "DON")Then Print"<script>alert(`You are not authorized to view Supplier documents Please contact your procurement administrator if you have questions`);` Print "history.go(-3)</script>" Exit Sub End If //Begin printing out HTML from the agent Print "<link rel=stylesheet type=""text/css"" href="""/transform /reqcat/css/default_styles.css"">" Dim db2c As New db2config (session) //Connect to DB2 src.database = db2c.getDB2DatabaseName() ` "reqcat41" src.Connect //build the SQL Query sqlQueryString = "select SUNAME, SUID from RC.SUPPLIER"
```

```

sqlQueryString = sqlQueryString & "order by SUNAME" If (src.execute(sqlQueryString, fldLst) = 0) Then Print "You do not have any supplier profiles to view." End End If Set suppname = fldLst.Lookup ("SUNAME") Set suppcode = fldLst.Lookup ("SUID") //Print more HTML Print "<TABLE width='100%' cellpadding='0' cellspacing='0'>" Print "<BR><TR><TD class='banner'" bgcolor="#3366cc" align=""center""> Suppliers</TD></TR>" Print "</TABLE>" //If the user is an admin, then provide a button for adding new suppliers If Admin Then Print "<form>" Print "<input type='button' name='mybutton' value='Add Supplier Profile' onClick='javascript:document.location.href= './SUPPLIER?OpenForm'; ''>" Print "</form>" Else Print "<BR>" End If Print "<table cellpadding=2 cellspacing=0 border=0>" Print "<tr><td class='bannersublevel'" ><B> Supplier</B></td>" Print "</tr>" //Toggling background color on alternate rows flag = True While (src.Fetch (fldLst) > 0) If (flag) Then Print "<tr bgcolor=#CCCCCC>" flag = False Else Print "<tr bgcolor=#FFFFFF>" flag = True End If //If user is an admin, then print the supplier name as a hotlink If Admin Then Print "<td class='field'"> <a href='./SUPPLIER? OpenForm&' & suppcode.text(0) & ">&suppname.text(0) &"</a> </td>" Else //otherwise just print it as text Print "<td class='field'"> &suppname.text(0)&"</td>" End If Print "<td class='field'"> &suppcode.text(0)&"</td>" Print "</tr>" Wend Print "</table>" End errHandler: Print "<br>" & session.currentagent.name & " - Line # " &Str(Erl) & " // Error " & Str(Err) & ":" & Error$ If (lclsxSession.Status <> LCSUCCESS) Then Dim text As String Dim extcode As Long Dim exttext As String Call lclsxSession.GetStatus (text, extcode, exttext) If (lclsxSession.Status = LCFAIL_EXTERNAL) Then Print "<br>DB2 message: " & exttext & " code #" & Cstr(extcode) Else Print "<br>Connector message: " & text End If Else Print Error$ End If End Sub

```

Detailed Description Paragraph Table (6):

```

TABLE 6 MENU BAR CODE ---- JavaScript (appNav.js) ----- // appNav Class Constructor -----
----- // This class implements a JavaScript Object intended to represent the Navigator function appNav(menuVar) { //methods this.init = appNavInit; this.sizeit = appNavSizeit; this.toggleText = appNavToggleText; this.isItem = appNavIsItem; this.reverse = appNavReverse; // properties this.ns = document.layers; this.ie = document.all; this.loaded = 0; this.whichone = 0; this.whichgroup = 0; if (this.ns) { this.show = `show`; this.hide = `hide`; } else { this.show = `visible`; this.hide = `hidden`; } this.menus = menuVar; this.max = menuVar.length; this.images = new Array(this.max); this.menuMove = new Array(this.max); for (i=0;i < this.max; i++) this.menuMove[i] = 20 * (menuVar[i] - 1); this.tabShow = [false, false, false, false, false, false, false, false]; } function appNavInit() { var k=0; if (this.loaded == 0) { for (i=0; i < this.menuMove.length; i++) { this.images[i] = new Array(this.menus[i]); for (j=0; j < this.menus[i]; j++) { this.images[i][j] = new Image(); pos = ("0"+i).slice(i>9,2) + ("0"+j).slice(j>9); this.images[i][j].src = "../images/men" + pos + ".gif" if (this.ie) { document.images[k].src=this.images[i][j].src; k++; } else document.layers["D"+pos].document.images[0].src=this.images[i][j].src; } } this.loaded=1; } }; function appNavSizeit() { if (this.loaded==1) {this.loaded = 0; this.init ()}; function appNavToggleText(z) { if (this.loaded==1) { this.tabShow[z] = !this.tabShow[z] for (j=1; j<this.menus[z]; j++) { pos = ("0"+z).slice(z>9) + ("0"+j).slice(j>9); if (this.ie) text = document.all("D"+pos).style else text = document.layers["D"+pos].style if (this.tabShow[z]) text.visibility = this.show; else text.visibility = this.hide; } for (i=z+1; i<this.max; i++) { for (j=0; j<this.menus[i]; j++) { pos = ("0"+i).slice(i>9) + ("0"+j).slice(j>9); if (this.ie) { text = document.all("D"+pos).style if (this.tabShow[z]) text.pixelTop += this.menuMove[z] else text.pixelTop -= this.menuMove[z] } else { text = document.layers["D"+pos].style if (this.tabShow[z]) text.top += this.menuMove[z] else text.top -= this.menuMove[z] } } } } } };

```

CLAIMS:

13. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for accessing from at least two disparate databases having different access control requirements code and data implementing a requisition catalog application on the web, said method steps comprising: populating in a first of said databases a relational database role table with user web identifiers and associated roles and levels selected from a plurality of possible roles and a plurality of possible levels for each said identifier; granting access to a database server responsive to request from a user having a user web identifier to locate in said role table said user's web identifier the role and level for said user; granting access to said database server to access in a second of said databases said code and data for said user; and responsive to said database server

accessing, executing code implementing said requisition catalog application with respect to said data to perform procedures authorized to said role and level for said user.

14. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for controlling access to code and tables in a first database and to data tables in a second database, said first database and said second database being of disparate types and having different access control requirements, said method steps comprising: providing in said first database a role table for each of a plurality of user web identifiers corresponding to role indicia selected from among a plurality of different role indicia and level indicia selected from among a plurality of different level indicia; providing access control list control to said code and tables in said first database; granting access to said role table to a server to obtain said role indicia and level indicia for said user; and operating said server to access both said first and second databases and to apply said role indicia and level indicia to access said code, said tables, and said data tables.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)